
Rule WLM113: BTE Phase samples were significantly less than total calculated samples

Finding: CPExpert has detected that the number of begin_to_end (BTE) phase samples recorded in the SMF Type 72 records were less than the total number of samples that would be collected based upon the transaction elapsed time. This finding applies only to service classes representing transactions under CICS/ESA Version 4 or later versions of CICS.

Impact: This finding means that long-running or never-ending transactions processed were in the service class. The presence of these transactions can distort response time calculations, particularly with standard reports produced by RMF.

Logic flow: The following rules cause this rule to be invoked:

- Rule WLM104: Subsystem Service Class did not achieve average response goal
- Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: CICS/ESA Version 4.1 (or later versions) reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*¹.

- **Begin_to_end phase.** The begin_to_end phase starts when CICS has classified the transaction². This action normally is done in a CICS Terminal Owning Region (TOR).
- **Execution phase.** The execution phase starts when either CICS or IMS (Version 5 or later) has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR). For IMS, this is done in an IMS Message Processing Region (MPR).

Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. These CICS transactions are termed "non-routed" transactions.

¹IMS Version 5 reports only *execution phase* samples.

²Classifying the transaction into a service class is done by the Workload Manager when the subsystem manager issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

CICS provides the System Resources Manager (SRM) with information about the phase (begin_to_end or execution) of transactions by executing the IWMMINIT ("Initialize the Monitoring Environment") macro. The DURATION parameter of the IWMMINIT macro tells the SRM whether the following information related to a transaction is associated with the begin_to_end phase or with the execution phase.

The IWMMINIT macro is issued immediately after CICS has issued the IWMCLSFY ("Assigning Incoming Work Requests to a Service Class") macro to establish a service class for a transaction. Thus, the SRM quickly knows (1) the service class to which a transaction belongs and (2) whether the transaction is in its begin_to_end phase or in its execution phase.

CICS or IMS will provide the SRM with information about the state of the transaction (active state, ready state, waiting state, etc.) by issuing the IWMMCHST ("Change State of Work Request") macro. The SRM simply sets bits in a status word to indicate the state of a transaction.

The SRM periodically samples the status word associated with each transaction³, and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class

The SRM also keeps a count of the number of samples that it takes of the begin_to_end phase and of the execution phase.

The counts of various samples are recorded in the "Work Manager/Resource Manager State Section" of SMF Type 72 records.

The SRM also includes the elapsed time of transactions (R723CTET) and the count of transactions (R723CRCP) in the SMF Type 72 records. Based on the transaction elapsed time and transaction count, CPExpert can compute the approximate number of samples that the SRM **should** take of the begin_to_end phase of transactions. Comparing the results of this computation against the actual number of begin_to_end samples reveals valuable information about the nature of the transactions.

³With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

To illustrate the computation, suppose that a single transaction were to execute in a service class, and further suppose that the transaction elapsed time was 1 second. During this second of elapsed time, the SRM should take a sample every 250 milliseconds (4 samples per second), or 4 samples of the begin_to_end phase⁴ of the transaction of the 1-second transaction. If two transactions with individual elapsed times of 1 second were to execute in the service class, the SRM should take 8 samples (1 second average elapsed time * 2 transactions * 4 samples per second = 8).

Thus, the computation of the number of samples that the SRM **should** take in any RMF measurement interval is simply the total elapsed time of transactions, times the sampling rate. The result from this computation should never be less than the number of samples that the SRM took of the begin_to_end phase, since the begin_to_end phase does not start until after the transaction has entered the system and has been classified to a service class, and the begin_to_end phase ends before the transaction is finally marked "ended" by the SRM.

Unfortunately, the result of the computation sometimes results in the number of begin_to_end phase samples being larger than the samples the SRM should take based on the elapsed time of transactions. This situation can occur when never-ending or long-running transactions execute in the service class.

The SRM updates the elapsed time of transactions only when the transactions end. Suppose that a never-ending transaction executed in the service class. The SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase). However, the SRM would never see the transaction complete and thus would not update the elapsed time of the transaction.

A similar situation occurs with long-running transactions. These transactions can span RMF measurement intervals; the SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase) in one RMF interval. The elapsed time of the transaction might not be recorded until a subsequent RMF interval.

These anomalies can cause response time calculations to be misleading, as discussed in Section 4. More importantly, the Workload Manager algorithms may be less effective if never-ending or long-running transactions are in the same service class as interactive transactions. This

⁴For the moment, we can ignore the time required by the SRM to assign the transaction to a CICS region, the time for the CICS region to issue the IWMCLSFY macro, the time for the Workload Manager to classify the transaction to a service class, and the time for the CICS TOR to issue the IWWMINIT macro. These times normally are very small.

is because the Workload Manager's computation of response times may be distorted by the long-running transactions.

CPEXpert can identify situations when the computed number of samples is significantly different from the expected number of samples. If the begin_to_end phase sample count is significantly less than the computed number of samples, CPEXpert can confidently conclude that there were long-running transactions executing in the service class, and that the long-running transactions ended in the RMF measurement interval being analyzed.

CPEXpert produces Rule WLM113 when the number of begin_to_end samples is less than 50% of the number of computed samples⁵. Rule WLM113 advises you that long-running transactions executed in the service class for which you have specified a response goal.

Suggestion: CPEXpert suggests that you identify the long-running transactions and remove them for the service class identified by Rule WLM110. Since the CICS transactions are long-running, it makes no sense to have the transactions in a service class with an interactive response goal.

IBM suggests the following guidance for CICS transactions:

- Do not mix CICS-supplied transactions with user transactions
- Do not mix routed with non-routed transactions
- Do not mix conversational with pseudo-conversational transactions
- Do not mix long-running and short-running transactions.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

⁵Actually, the 50% value may be overly generous. As mentioned in Footnote 4, the delay between the time MVS recognizes a transaction and the begin_to_end phase begins should be quite small. It is possible that CPEXpert may use some much larger value (e.g., 90%) in the future. On the other hand, long-running transactions will likely have a **significant** effect on analysis and probably will be identified with the 50% value.

CICS/TS Release 1.3 Performance Guide
Section 2.5.7.1: Service Definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions). |